From Hidden to Recognized: Direct Decoding of Named Entities from Sparse Autoencoder Features in Large Language Models

Anonymous ACL submission

Abstract

Large Language Models (LLMs) are increasingly used for Named Entity Recognition (NER) and synthetic data generation, yet their label annotation processes remain largely black boxes. This lack of transparency hinders reliability and control in LLM-based annotation pipelines. To address this, we investigate whether Sparse Autoencoders (SAEs) can extract interpretable features from LLM activations to decode named entities directly. Evaluating on general and biomedical NER datasets, we show that SAEs effectively capture entity-relevant features, outperforming standard probing classifiers in the biomedical domain. Our findings suggest that SAEs offer a promising step toward more transparent and controllable LLM-based annotation and synthetic data generation pipelines.

1 Introduction

Generating high-quality labeled data is a persistent challenge for Information Extraction (IE) tasks like Named Entity Recognition (NER), particularly in low-resource domains. Large Language Models (LLMs) offer a promising way for generating synthetic labeled data to address this scarcity (Josifoski et al., 2023; Hiebel et al., 2023; Vuth et al., 2024). However, these works that use LLMs for annotation often treat the model as a black box. This lack of transparency hinders understanding *why* specific annotations are produced, limiting the controllability needed for robust synthetic data generation pipelines. To improve the reliability and interpretability of LLM-based annotation for structured tasks, there is a critical need for methods that provide insight into the LLM's internal processing of relevant linguistic and semantic features.

Mechanistic interpretability methods offer a way to open the "black box" of LLMs and understand the specific features they learn. Sparse Autoencoder (SAE) is a promising technique that decomposes dense neural activations into a sparse set of interpretable, disentangled latent features or "monosemantic neurons" (Bricken et al., 2023; Cunningham et al., 2023; Gao et al., 2024; Rajamanoharan et al., 2024). Recent works have shown that SAE representations are useful for downstream tasks, such as training classifiers for privacy-risk detection (Bricken et al., 2024; Gallifant et al., 2025). However, the potential of leveraging the learned *latent features* for understanding and performing *structured prediction* tasks like NER directly, without training complex dense classifiers, is not well-explored. Validating whether SAEs can capture and provide direct access to entity-specific features within LLMs is a crucial step towards building more transparent and analyzable annotation processes, including those used in synthetic data generation.

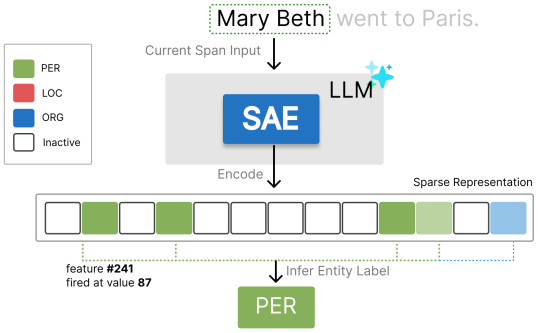

Figure 1: Visualization of the heuristic for decoding named entity labels via Sparse Autoencoder.

This paper investigates whether named entities can be directly decoded from the interpretable latent features learned by an SAE applied to LLM hidden representations. We evaluate this approach on general-domain and biomedical NER datasets, demonstrating that SAEs effectively capture entity-relevant features. Our findings show that decoding entity labels directly from these interpretable features is competitive with standard probing classi-

fiers based on dense representations, and can even outperform them in biomedical contexts. This research highlights the potential of interpretable features for label-efficient IE and provides foundational evidence for developing future LLM-based annotation methods grounded in model understanding.

2 Related Works

Utilizing the rich information within LLM hidden representations for downstream tasks like NER is a common technique. Probing classifiers (Tenney et al., 2019; Hewitt and Manning, 2019; Bommasani et al., 2021) are a widely adopted method, training task-specific models on dense LLM representations. While effective for achieving performance on a task, these methods operate on high-dimensional, opaque vectors and train probe classifiers, providing limited interpretability into the specific features driving predictions. Moreover, the efficacy of probing as a diagnostic tool remains debated, partly because its performance is highly dependent on the chosen classifier, making it difficult to isolate and assess the quality of the underlying representation itself (Pimentel et al., 2020; Belinkov, 2021; Zhou and Srikumar, 2021).

Mechanistic interpretability aims to provide a more transparent view into LLM internals. SAE is a promising technique in this field, designed to decompose dense activations into a sparse set of disentangled and human-interpretable latent features. While SAEs have shown promise in revealing specific concepts encoded in LLMs within classification tasks like privacy-risk detection (Bricken et al., 2024; Gallifant et al., 2025), their potential for directly decoding information relevant to structured prediction tasks like NER from these interpretable features remains less explored. Our work investigates this specific application of SAEs, detailed in the following sections.

3 Sparse Autoencoders (SAE)

Neurons in deep neural networks are often not inherently interpretable. Many are *polysemantic*, they respond to a mixture of seemingly unrelated features. For example, Bricken et al. (2023) found that a single neuron can simultaneously activate for academic concepts, English dialogue, HTTP requests, and Korean text. A leading explanation for this phenomenon is *superposition* (Elhage et al., 2022), where a model encodes more features than it has neurons by representing each feature as a unique linear combination of neurons.

SAE aims to resolve this issue by extracting more interpretable, *monosemantic* components from a model's internal representations. By enforcing sparsity constraints, SAE encourages each feature to activate in isolation, therefore disentangling polysemanticity in the hidden space. Intuitively, SAE decodes what each neuron might represent by first decomposing the input representation into sparse features and then reconstructing it. For instance, given a 100-dimensional activation vector from a transformer's MLP layer, we can train an SAE to map this to a 256-dimensional hidden representation with enforced sparsity, e.g., allowing only 15 nonzero elements. This constraint forces the model to discover interpretable features that correspond more closely to human-understandable concepts.

Concretely, SAE operates on hidden representations (e.g., from an MLP layer L^l_{MLP})[1] using a pair of encoder and decoder functions:

$$f(\mathbf{x}) := \sigma(W_{\text{enc}}\mathbf{x} + \mathbf{b}_{\text{enc}})$$

$$\hat{\mathbf{x}} := W_{\text{dec}} f(\mathbf{x}) + \mathbf{b}_{\text{dec}}$$

Here, $\mathbf{x} \in \mathbb{R}^n$ is the original activation vector, $f(\mathbf{x})$ is the sparse encoded representation, and $\hat{\mathbf{x}}$ is the reconstruction. $W_{\text{enc}} \in \mathbb{R}^{m \times n}$ and $W_{\text{dec}} \in \mathbb{R}^{n \times m}$ are the encoder and decoder weight matrices, respectively. Each column d_i of W_{dec} represents a learned direction or *feature dictionary* component into which $\mathbf{x}$ is decomposed. The activation function σ enforces non-negativity and sparsity. Different activation functions σ have been used for training SAEs, such as ReLU (Bricken et al., 2023; Cunningham et al., 2023), TopK activation (Gao et al., 2024) and JumpReLU (Rajamanoharan et al., 2024).

4 Experiment Setup

We use pre-trained SAE models released by Lieberum et al. (2024), which were trained on the hidden activations of `Gemma 2-9B-IT`.[2] Specifically, we experiment with SAEs extracted from three transformer layers: $l \in 9, 20, 31$, and two feature dictionary sizes: $d \in 16k, 131k$. We denote each configuration as S^d_l. For evaluation, we use

[1] Activations may also be cached from attention layers or residual streams.

[2] SAEs for the Gemma 2 model family are available at: https://huggingface.co/google/gemma-scope

two datasets from different domains: Re-DocRED (Tan et al., 2022) from the general domain, and AnEM (Ohta et al., 2012) from the biomedical domain. As baselines, we train simple one-hidden-layer probing classifiers (ReLU, 512 units) on hidden activations from the base model's middle (layers 10–20) and final residual layers, repeating each experiment five times.

4.1 Feature to Label Dictionary

Before inferring entity labels from sparse representations, it is crucial to first identify which features are most relevant to each entity type. A straightforward approach is to use the automatically generated *auto-interp*[3] descriptions of each SAE feature and compare them against textual descriptions of the target entity types using cosine similarity.

However, this approach often leads to ambiguous matches, as the LLM-generated descriptions are broad or overlapping. For instance, with SAE model S_{31}^{16k}, we observed the following cases:

- Feature **8877**: "reference to singular specific entities or specific items" $\xrightarrow{label}$ [MISC]
- Feature **11101**: "reference to specific individuals or notable events" $\xrightarrow{label}$ [PER, EVENT]
- Feature **713**: "names and references related to people, locations, and organizations" $\xrightarrow{label}$ [PER, LOC, ORG]

To address this ambiguity, we propose a data-driven heuristic to cluster relevant features based on their activations across labeled data.

Feature Filtering Heuristic. The core intuition behind our method is to identify features that consistently activate over specific entity spans by analyzing their statistical association with labeled entity types. We run the SAE on a training dataset and cache feature activations for each entity span. For each feature, we compute its activation frequency across entity types and assess significance using a Chi-square test:

$$\chi^2 = \sum_{e \in E} \sum_{a \in A} \frac{(O_{e,a} - P_{e,a})^2}{P_{e,a}} \quad (1)$$

Here, E is the set of entity types, and $A = \{\text{activated}, \text{not activated}\}$ denotes the activation status. $O_{e,a}$ is the observed frequency of activation a for entity type e, and $P_{e,a}$ is the expected frequency under the null hypothesis of independence. The heuristic is shown in Algorithm 1.

Algorithm 1: Feature Filtering

Input: SAE activations $\mathcal{Y}$, labeled dataset $\mathcal{D}$, entity types E, significance threshold α
Output: Feature-to-Label Dictionary

```
Cache activations from 𝒴 for labeled spans in 𝒟;
Compute activation counts per feature f and entity label e ∈ E;
foreach feature f do
    Compute Chi-square statistic χ² Eq. 1;
    Compute p-value from Chi-square distribution;
    if p < α then
        Assign feature f to entity label with highest O_{e,activated};
    end
end
return Feature-to-Label Dictionary;
```

4.2 Inference

We used `SAELens` (Joseph Bloom and Chanin, 2024) to cache token-level feature activations from different SAEs, S_l^d mentioned in Section 4. Specifically, we selected SAEs trained on the **residual stream** of `Gemma 2-9B-IT` with an average L_0 sparsity closest to 100. Since SAE activations are token-level representations, an aggregation strategy is necessary to obtain span-level features. While recent work Gallifant et al. (2025) employed top-N activated features per token followed by binarization, we adopt a simpler approach: **sum-pooling** activations across all tokens within the span. This results in a fixed-size feature vector representing the entire entity span. To infer entity labels from these aggregated span representations, we compute a probability distribution over entity types using a softmax applied directly to the sum of relevant feature activations:

$$P(e \mid \phi) = \frac{\exp\left(\sum_{f \in F_e} \phi(f)\right)}{\sum_{e' \in E} \exp\left(\sum_{f \in F_{e'}} \phi(f)\right)} \quad (2)$$

where F_e is the set of features associated with entity type e, filtered using the heuristic described in

[3]Each SAE feature is paired with a natural language description generated by a larger LLM.

		LOC	MISC	NUM	ORG	PER	TIME	Macro-F1	Weighted-F1
Layer 9	16k	0.743	0.481	0.929	0.333	0.867	0.893	0.708	0.701
	131k	0.765	0.540	0.910	0.444	0.857	0.889	**0.734**	**0.729**
Layer 20	16k	0.599	0.218	0.719	0.116	0.572	0.869	0.516	0.516
	131k	0.592	0.256	0.654	0.209	0.661	0.681	0.509	0.516
Layer 31	16k	0.724	0.541	0.770	0.241	0.846	0.915	0.673	0.682
	131k	0.697	0.476	0.751	0.206	0.833	0.887	0.642	0.651
Probing	Mid	0.821	0.585	0.800	0.635	0.865	0.853	0.760	0.771
	Last	0.795	0.507	0.786	0.599	0.784	0.842	0.719	0.729

Table 1: F1-scores across layers and dictionary sizes on Re-DocRED dataset.

		Anatomical System	Cell	Cellular Component	Developing Anatomical	Immaterial Anatomical	Multi-tissue Structure	Organ	Organism Subdivision	Organism Substance	Pathological Formation	Tissue	Macro F1	Weighted F1
Layer 9	16k	0.000	0.620	0.296	0.213	0.000	0.631	0.439	0.133	0.786	0.735	0.000	0.352	0.526
	131k	0.353	0.733	0.328	0.766	0.125	0.627	0.641	0.129	0.786	0.708	0.421	**0.511**	**0.614**
Layer 20	16k	0.000	0.636	0.065	0.000	0.000	0.509	0.103	0.035	0.740	0.467	0.000	0.232	0.401
	131k	0.000	0.662	0.329	0.160	0.000	0.614	0.644	0.069	0.758	0.637	0.209	0.372	0.554
Layer 31	16k	0.000	0.774	0.390	0.160	0.176	0.622	0.693	0.423	0.796	0.786	0.282	0.464	0.637
	131k	0.000	0.803	0.390	0.160	0.278	0.593	0.601	0.119	0.751	0.733	0.395	0.442	0.615
Probing	Mid	0.422	0.442	0.317	0.337	0.252	0.309	0.699	0.478	0.652	0.293	0.315	0.409	0.427
	Last	0.317	0.348	0.185	0.350	0.169	0.285	0.560	0.357	0.621	0.233	0.300	0.338	0.358

Table 2: F1-scores across layers and dictionary sizes on AnEM dataset.

Algorithm 1, and $\phi(f)$ denotes the activation value of feature f over the span. The final predicted entity type is defined as:

$$\hat{y} = \arg\max_{e \in E} P(e \mid \phi) \tag{3}$$

4.3 Results and Discussion

The results of our experiments are presented in Table 1 and Table 2. We observe that using feature activations from layer 9 consistently outperforms layers 20 and 31. This contrasts with previous findings, where feature activations from layer 20 were reported to yield the best performance (when used for training a classifier) (Gallifant et al., 2025). Our findings suggest that early layers (e.g., layer 9) may retain more generalizable and disentangled features, which are beneficial for NER. We also observed different behaviors with respect to SAE width. On the general-domain dataset Re-DocRED, the 16k-width SAE tends to yield better results, whereas the opposite is true for the biomedical-domain dataset AnEM. We attribute this to the phenomenon of *feature splitting* (Bricken et al., 2023), where narrower SAEs tend to decompose features into more specialized ones when scaled up. These specialized features benefit closed-domain datasets like AnEM, which require finer-grained features that activate on niche token spans. Across both datasets, hidden activations from middle layers tend to yield better performance when used to train probing classifiers. Notably, predictions from SAE layer 9 remain competitive with those of trained probes on Re-DocRED. However, these predictions struggle with ambiguous classes such as `MISC` and `ORG`. Our error analysis indicates that this is due to overlapping or ambiguous features between classes like `LOC`, `MISC`, and `ORG`. Further refinement of our filtering heuristic is needed to resolve this. While on AnEM, SAE predictions from layer 9 significantly outperform probing classifiers. These observations reinforce that different classifiers may be needed to reach optimal performance (Zhou and Srikumar, 2021). While probing classifiers can learn to make accurate predictions when given sufficient data (as seen with Re-DocRED), direct decoding from interpretable SAE features demonstrates advantages where the learned feature space is well-aligned with the task, particularly in specialized domains.

5 Conclusion

This work demonstrates that named entities can be directly decoded from interpretable latent features learned by SAE. Evaluating this approach, we found performance competitive with probing classifiers in general domains and outperformed them in biomedical contexts. Our findings highlight the utility of interpretable features for transparent Information Extraction, supporting the development of more reliable LLM-based annotation and synthetic data generation pipelines.

Limitations

This study has several limitations that point towards important paths for future work.

- Our experiments are limited to SAEs trained on a specific model (`Gemma 2-9B-IT`) and solely using residual stream activations. Future research should explore other LLM architectures and different activation sources (e.g., MLP, Attention layers) to assess the generalizability of our findings.
- The current method relies on access to ground-truth entity offsets to extract and cache activations for relevant spans. This limits its direct applicability in real-world scenarios. A crucial direction is developing heuristics or models for automatic detection of entity mention spans. One potential heuristic is to leverage constituent parsing to identify syntactic boundaries aligned with entity mentions.
- Our analysis is currently restricted to English. Future work should extend this analysis to multilingual settings to understand how well SAE representations generalize across languages, particularly for potential impact in low-resource languages.

Addressing these limitations could pave the way for developing fully automated, interpretable annotation pipelines using SAE features, which would be especially valuable for generating high-quality synthetic data for low-resource domains.

References

Yonatan Belinkov. 2021. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48:207–219.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, S. Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen A. Creel, Jared Davis, Dora Demszky, Chris Donahue, Moussa Koulako Bala Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas F. Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, O. Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Benjamin Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, J. F. Nyarko, Giray Ogut, Laurel J. Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Robert Reich, Hongyu Ren, Frieda Rong, Yusuf H. Roohani, Camilo Ruiz, Jack Ryan, Christopher R'e, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishna Parasuram Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei A. Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2021. On the opportunities and risks of foundation models. *ArXiv*, abs/2108.07258.

Trenton Bricken, Jonathan Marcus, Siddharth Mishra-Sharma, Meg Tong, Ethan Perez, Mrinank Sharma, Kelley Rivoire, and Thomas Henighan. 2024. Using dictionary learning features as classifiers. `https://transformer-circuits.pub/2024/features-as-classifiers/index.html`. Accessed: 2025-04-17.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023. Sparse autoencoders find highly interpretable features in language models. *ArXiv*, abs/2309.08600.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. 2022. Toy models of superposition. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2022/toy$_m odel/index.html$.

Jack Gallifant, Shan Chen, Kuleen Sasse, Hugo Aerts, Thomas Hartvigsen, and Danielle S. Bitterman. 2025. Sparse autoencoder features for classifications and transferability. *ArXiv*, abs/2502.11367.

Leo Gao, Tom Dupr'e la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. 2024. Scaling and evaluating sparse autoencoders. *ArXiv*, abs/2406.04093.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *North American Chapter of the Association for Computational Linguistics*.

Nicolas Hiebel, Olivier Ferret, Karen Fort, and Aurélie Névéol. 2023. Can synthetic text help clinical named entity recognition? a study of electronic health records in French. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2320–2338, Dubrovnik, Croatia. Association for Computational Linguistics.

Curt Tigges Joseph Bloom and David Chanin. 2024. Saelens. `https://github.com/jbloomAus/SAELens`.

Martin Josifoski, Marija Sakota, Maxime Peyrard, and Robert West. 2023. Exploiting asymmetry for synthetic training data generation: SynthIE and the case of information extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1555–1574, Singapore. Association for Computational Linguistics.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, J'anos Kram'ar, Anca Dragan, Rohin Shah, and Neel Nanda. 2024. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. *ArXiv*, abs/2408.05147.

Tomoko Ohta, Sampo Pyysalo, Jun'ichi Tsujii, and Sophia Ananiadou. 2012. Open-domain anatomical entity mention detection. In *Proceedings of the Workshop on Detecting Structure in Scholarly Discourse*, pages 27–36, Jeju Island, Korea. Association for Computational Linguistics.

Tiago Pimentel, Naomi Saphra, Adina Williams, and Ryan Cotterell. 2020. Pareto probing: Trading-off accuracy and complexity. *ArXiv*, abs/2010.02180.

Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, János Kramár, and Neel Nanda. 2024. Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders. *ArXiv*, abs/2407.14435.

Qingyu Tan, Lu Xu, Lidong Bing, Hwee Tou Ng, and Sharifah Mahani Aljunied. 2022. Revisiting docred – addressing the false negative problem in relation extraction. In *Proceedings of EMNLP*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. Bert rediscovers the classical nlp pipeline. In *Annual Meeting of the Association for Computational Linguistics*.

Nakanyseth Vuth, Gilles Sérasset, and Didier Schwab. 2024. KGAST: From knowledge graphs to annotated synthetic texts. In *Proceedings of the 1st Workshop on Knowledge Graphs and Large Language Models (KaLLM 2024)*, pages 43–55, Bangkok, Thailand. Association for Computational Linguistics.

Yichu Zhou and Vivek Srikumar. 2021. Directprobe: Studying representations without classifiers. *ArXiv*, abs/2104.05904.